PathMakeUniqueName

Generated pathname easily guessed

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 9489 bytes

Attack Category	• Path spoofing or confusion problem
Vulnerability Category	• Indeterminate File/Path • TOCTOU - Time of Check, Time of Use • Temporary file creation problem
Software Context	• Filename Management • Temporary File Management
Location	• shlobj.h
Description	Files created with names generated by PathMakeUniqueName() may not be secure. Names created by PathMakeUniqueName() may be easily guessed by an attacker, allowing the attacker to gain access to the file and its data. These functions basically append a digit to the end of the filename to make it unique. For example, "My file" is changed to "My file (2)". PathYetAnotherMakeUniqueName is vulnerable in the same manner, with a couple of different parameters for long/short filenames.
APIs	

Function Name	**Comments**
PathMakeUniqueName	check
PathYetAnotherMakeUniqueName	check

Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results. These routines create filenames that are trivially easy for an attacker to guess, which allows for a

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	race condition to pre-create the file with attacker-specific access permissions. For example, the routine will check for the template filename "MyFile" and determine that the next filename in sequence is "MyFile (1)". After this check, the attacker could create the file with his own set of ownership and permissions of that file prior to the program being able to create and use the file.
Exception Criteria	
Solutions	

Solution Applicability	**Solution Description**	**Solution Efficacy**
Generally applicable to all pathmakeuniquename() calls.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.
Generally applicable to all pathmakeuniquename() calls.	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
Generally applicable to all pathmakeuniquename() calls.	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
Generally applicable to all	Recheck the resource after	Effective in some cases.

pathmakeuniquename() calls.	these call to verify that the action was taken appropriately.	
When naming a file that needs to be secure.	Never use pathmakeuniquename() for creating secure temporary files.	Effective.
Whenever a temporary file is needed.	Follow these guidelines to minimize security risk: First, temporary files should be created in a secure directory. Then, follow these steps to create the file: • Pick a prefix for the filename (e.g., /tmp/my_pref) • Generate at least 64 bits of high-quality cryptographical randomness • base64 encode the random data (substitute "." for "/") • Concatenate the prefix with the random data • set umask apprpropriately (0066 is usually good)	Effective if all recommendations are followed.

		• Use fopen() to create the file in the proper mode • delete the file immediately using unlink() • NOTE: If tmp file is on network drive (not recommended), cannot do this. • Perform reads, writes, etc. on file descriptor. • close the file. Automatically deleted already. NEVER close and re-open the file if it lives in a directory that may be succeptible to a race condition. NOTE: It's possible to have a race condition between the fopen() and unlink() commands. Doing all this in a secure directory minimizes that risk.	
	Whenever the file should be secure but not be temporary.	Avoid the unlink() step recommended in the above	Effective.

	solution for temporary files.	
Whenever filename needs to be related to original filename.	Use file name as a prefix and add a cryptographic-quality random suffix	Effective.
When naming a file that needs to be secure.	Do all work in a secure directory.	Effective.

Signature Details	BOOL PathMakeUniqueName(LPWSTR pszUniqueName, UINT cchMax, LPCWSTR pszTemplate, LPCWSTR pszLongPlate, LPCWSTR pszDir);

Examples of Incorrect Code	`WCHAR uniqueName[MAX_PATH];`
	`LPWSTR pszUniqueName = uniqueName;`
	`if (! PathMakeUniqueName(pszUniqueName, L"AShortNm.txt", L"A Long Name.txt", L"C:\\SomeDirectory"))`
	`{`
	`handleError();`
	`}`

Examples of Corrected Code	`const int numberRandBytes = 16;`
	`char randBytes[numberRandBytes];`
	`const int sizeRandChars = 2*numberRandBytes+1; // size is conservative upper bound`
	`char randChars[sizeRandChars]; // buffer for rand characters`
	`const char myPrefix[] = " /tmp/ my_pref";`
	`char filePath[MAX_PATH];`
	
	`// Hypothetical routine to generate random data.`
	`// Should use a cryptographic toolkit rather than using random() or similar non-crypto functions,`
	`// if really want to make result hard to guess.`
	`createRandomData(randBytes, numberRandBytes );`
	
	`// Hypothetical base64 encoding routine that uses "." instead of "/"`

```
convertToModifiedBase64(randChars,
sizeRandChars, randBytes,
numberRandBytes);

strncpy(filePath, myPrefix, MAX_PATH);
strlcat(filePath, randChars, MAX_PATH);

// Should really check to ensure
that file does not already exist,
and set umask - omitted

FILE *tempFile;
tempFile = fopen(filePath, "w+");
unlink(filePath);
```

Source Reference	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/functions/pathmakeuniquename.asp[2]
Recommended Resource	
Discriminant Set	**Operating System**: • Windows; **Languages**: • C • C++

1. mailto:copyright@cigital.com